Institutionen för Kommunikation och Information
Examensarbete I datalogi med inriktning mot nätverks- och systemadministration, 15 HP

C-nivå

Vårterminen 2011

# SCTP

## An analysis of proposed implementations

**Mattias Hedén**

Mattias Hedén
b08mathe@student.his.se
19821008-5539

Datum (2011-06-05)

# SCTP – An analysis of proposed implementations

Submitted by Mattias Hedén to the University of Skövde as a final year project towards the degree of B.Sc. in the School of Humanities and Informatics. The project has been supervised by Rose-Mharie Åhlfeldt.

**2011-06-05**

I hereby certify that all material in this final year project which is not my own work has been identified and that no work is included for which a degree has already been conferred on me.

Signature: _____

**SCTP – An analysis of proposed implementations**

**Mattias Hedén**

# Abstract

There are several weaknesses in the popular transport protocol TCP (Transmission Control Protocol). A possible replacement to TCP would be the newer protocol SCTP (Steam Control Transmission Protocol). This thesis presents three different proposed implementations of SCTP: HTTP over SCTP, online games over SCTP and IP mobility over SCTP. The proposed implementations are analyzed, based on relevant literature, and recommendations are issued on the importance of moving forward with them.

The result of the thesis is that HTTP over SCTP is recommended. SCTP features such as multi-streaming, multi-homing and the four-way handshake addresses the inherent weaknesses with using TCP for HTTP traffic. IP mobility over SCTP is also recommended since it results in lower delay in the handover process compared to MIPv6 (Mobile IPv6). Online games over SCTP, however, is not recommended since the existing implementations of SCTP results in poor latency for the kind of traffic online games produce.

**Keywords:** SCTP, HTTP, online games, IP mobility

# Index

# 1 Introduction

Stream Control Transmission Protocol (SCTP) is the name of an end-to-end transport protocol which shares many properties with the two most common transport protocols in use: Transmission Control Protocol (TCP), which is used for reliable, connection-oriented traffic, and User Datagram Protocol (UDP), used for unreliable, connectionless traffic (Internet Engineering Task Force, 2002).

One might wonder why SCTP is needed considering the wide spread of TCP and UDP. The answer would be that there are certain weaknesses in TCP, such as head-of-line blocking and susceptibility to SYN-flooding attacks, which does not exist in SCTP. Furthermore SCTP supports several features that are not supported by either TCP or UDP such as multi-homing and multi-streaming (Internet Engineering Task Force, 2007a). The weaknesses of TCP and the features of SCTP will be further elaborated on in chapter 2.

Despite the fact that research regarding SCTP is very active and it seems like there is a need for SCTP, there are few implementations of the protocol. Thorstensson (2010), for example, questions why SCTP has not been more widely implemented since it would stop the harmful SYN-flooding attacks that TCP is susceptible to and which are hard to protect a system from. Considering that TCP is one of the foundations of the current Internet infrastructure and handles most traffic which requires reliability and connection-establishment it may be a valid question if one consider the magnitude of the weakness.

## 1.1 Motivation

Considering the strengths of SCTP and the weaknesses of TCP one might wonder why it has not seen wider use. One explanation is that the protocol is relatively new and unknown, TCP has been around for decades and the major weaknesses did not surface until after many years. Some worry that SCTP will also turn out to have new weaknesses which will take time to discover (Jones, 2006). Amer et al. (2006) however argues that SCTP is a better transport protocol for the web infrastructure due to the weaknesses of TCP and the new features of SCTP, in particularly the susceptibility to SYN-flood attacks and the negative effect of head-of-line blocking in TCP is cited. Labovitz et al. (2008) analyzed data from 67 ISP:s (Intenet Service Provider) which revealed that SYN-flood attacks stood for over 30% of all Distributed Denial-of-Service (DDoS) attacks. Knowledge of the protocol and the possible implementations is key to the process of increasing the cases where SCTP is actually implemented.

## 1.2 Aim

The aim of this thesis is to present various proposals for implementation of SCTP, analyze these proposals and arrive at a recommendation for how critical it is to implement them. By doing this it is hoped that this thesis will further stimulate research on SCTP and the implementation of the protocol in more applications.

## 1.3 Related works

Fu & Atiquzzaman (2004) performed a study where they used existing research to identify areas where SCTP could be of use, as well as some issues and challenges to be addressed before SCTP could be used in applications. It has been 7 years since then and the Internet Engineering Task Force (2007a) has since updated the standard for SCTP, removing several issues, and a lot of new research has taken place. In the University of Skövde's article database LibHub there is well over 200 new articles since 2004 with SCTP as keyword.

There are still some challenges to overcome, such as the lack of built-in support for SCTP in the Windows and Mac operative systems (FreeBSD, Linux and Solaris do have this support) for which users has to rely on third party packages. Amer & Stewart (2007) predicts an increased use of SCTP in applications and that eventually all operative systems will offer support for SCTP but that until that day many applications will hesitate to implement SCTP.

# 2 Background

To understand why it might be necessary to implement SCTP in certain cases, instead of relying on the older TCP protocol, one must have an understanding of the protocols, the weaknesses inherent in TCP and the new features available in SCTP. This chapter will begin with a short overview of TCP and its major weaknesses followed by an overview of SCTP and the major features available.

## 2.1 Transmission Control Protocol (TCP)

TCP is a connection-oriented transport protocol for reliable transmission of data between end-points in a network, it is widely in use in applications which requires such reliable transmissions (Internet Engineering Task Force, 1981).

Being connection-oriented means that to send data between two end-points they must first establish a TCP connection between them. This session is established by means of a method called a three-way handshake (As shown in Figure 1), the way this handshake works is that the party that wishes to initiate a connection will send a message with the SYN-flag set, which indicates the wish to set up a session, as well as a sequence number that identifies the message order for the session. The recipient then responds with a message where the SYN-flag is set as well as an ACK-flag which indicates acknowledgment of the request, the ACK-number will be set to the sequence number received incremented by 1 to indicate that it is ready to receive the next message, it will also send back a new sequence number to be used for messages sent in the other direction of the session. Finally the initiating party responds with a message that has the ACK-flag set and the ACK-number set to the received sequence number incremented by one as well as sequence number corresponding to the received ACK-number (Internet Engineering Task Force, 1981).

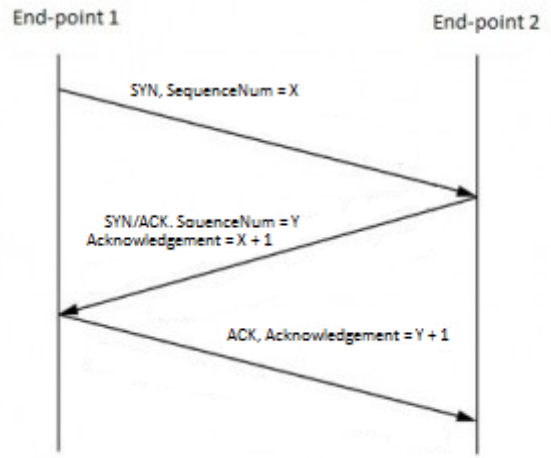


*Figure 1: This picture illustrates the process of establishing a session between two end-points.*

### 2.1.1 SYN-flooding

A SYN-flood attack is a form of denial-of-service (DoS) attack which specifically targets a weakness in the session establishment of the TCP protocol. This weakness comes from how the protocol handles packets which are lost or damaged. The end-point which receives the initial SYN request maintains a SYN_RECV queue for each port which keeps track of all the instances where a SYN/ACK has been sent out but where no ACK has yes been received. Attacker can abuse this by sending out multiple SYN requests while not answering the SYN/ACK responses. Eventually the queue will run out of space and further incoming SYN requests will be dropped effectively denying access to any services on that port. This works because such half open sessions are not dropped until a few minutes has passed because of the potential delays caused by routing over the Internet (Pfleeger & Pfleeger, 2006).

To avoid revealing their IP addresses and to make the SYN-flood packets harder to distinguish from normal traffic an attacker will often spoof the return address included in the SYN packet. Inspecting the queue would otherwise reveal the attackers identity. This is possible because using a nonexistent address results in an ICMP Destination Unreachable response which the TCP connection is not able to process (Pfleeger & Pfleeger, 2006).

### 2.1.2 Head-of-line blocking

TCP is a protocol which enforces strict ordering of data. Head-of-line blocking occurs when a packet is lost and the following packets arrives out of order. The packets that do arrive are held until the lost packet has been retransmitted and received. This ensures that all the data arrives and is processed in the correct order, but also means that a single lost packet will result in a delay for all subsequent packets even if they are not related to the packet which was lost. Since many independent messages can be sent over a TCP connection at the same time this leads to unnecessary delays for some of them (Smith, 2009).

## 2.2 Stream Control Transmission Protocol (SCTP)

SCTP is, much like TCP, a connection-oriented transport protocol for reliable transmission of data between two end-points. SCTP is a much more recent protocol than TCP, the first Request For Comments (RFC) for SCTP was released in 2000 and the protocol has yet to see much use. SCTP was designed with the weaknesses of TCP in mind and thus have several differences. For example session initialization is handled differently and SCTP has support for both Multi-homing and Multi-streaming (Internet Engineering Task Force, 2007a).

What differentiates the session initialization in SCTP from TCP is the use of a four-way handshake to set up an association rather than a three-way handshake as well as the use of a cookie. The party that wishes to initiate a session sends an INIT message which includes a verification tag in the initiate tag field, this tag is a random number between 1 and 2^32 -1, it wont change during the session and is used to verify the parties in the session. After having sent the INIT message the end-point will enter the COOKIE-WAIT state. Upon receiving the INIT message an INIT ACK response message will be sent. The verification tag will be set to the number in the initiation

tag field provided by the INIT message and a new random number will be generated and set in the initiation tag field to act as verification tag in the other direction.

At this stage it is important to note that there is no half-open session unlike in TCP, the party which received the INIT message will not save any information regarding the session to avoid susceptibility to resource based attacks such as SYN-flooding. Instead of saving information about the session a state cookie is generated with all the information needed for the session. This state cookie is included in INIT ACK message and after it has been sent any local resource related to the session will be deleted (Internet Engineering Task Force, 2007a).

The party which initiated the session will receive the INIT ACK message and leave the COOKIE-WAIT state. In response it will send a COOKIE ECHO message which contains the state cookie and then enter the COOKIE-ECHOED state. Included in the COOKIE ECHO message can also be any data that is waiting to be sent, any additional packets may however not be sent until a COOKIE ACK message is received. When the other end-point receives the COOKIE ECHO message it responds with a COOKIE ACK and moves the session to the ESTABLISHED state, like the COOKIE ECHO message there can be data included in the COOKIE ACK. Upon receiving the COOKIE ACK message will make the first end-point move the session to an ESTABLISHED state as well and the session will be fully initialized resulting in an association between the two end-points. Allowing for sending of data in these two steps reduces the time lag between the start of session initiation and data transfer that would otherwise occur in the lengthier four-way handshake (Internet Engineering Task Force, 2007a). Figure 2 shows a comparison between the session initialization process of SCTP and TCP.

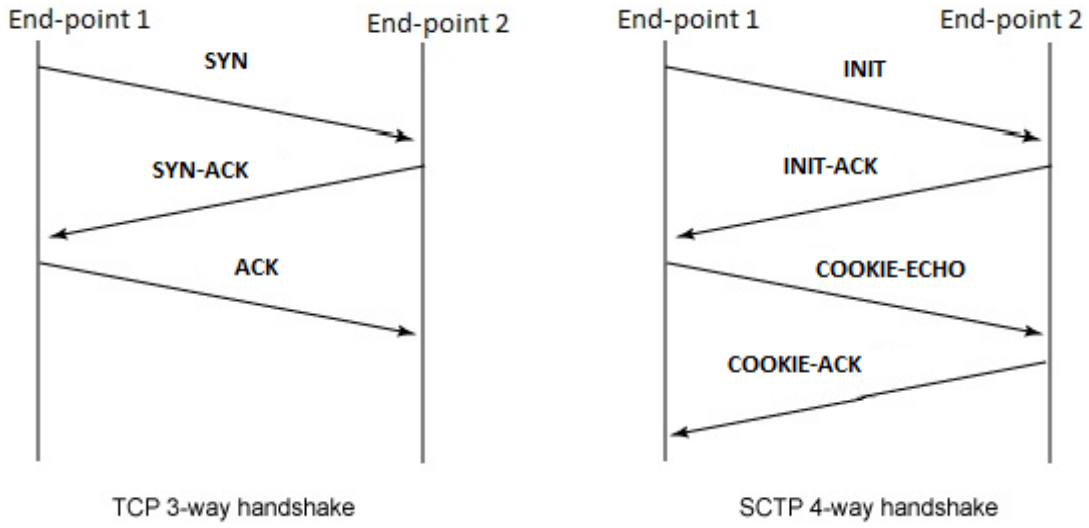


*Figure 2: These pictures show the different messages sent between end-points in order to initiate a session in TCP and SCTP.*

### 2.2.1 Multi-streaming

The multi-streaming feature is SCTP:s answer to head-of-line blocking. Since only having a single stream leads to unrelated messages being delayed in the case of a lost packet SCTP employs multiple streams within the same association. These streams provides separate sequences to provide ordered sending of messages that are independent of each other (Internet Engineering Task Force, 2007a). For example sending three files simultaneously over an SCTP association would result in three separate streams each of which would only be susceptible to delays due to packet loss in its own stream (see Figure 3). It is possible to achieve multi-streaming in TCP but it requires setting up a separate TCP connection for each stream. Using multiple connections results in unfair allocation of the available bandwidth since more connections would result in a larger share (Iqbal, 2003).

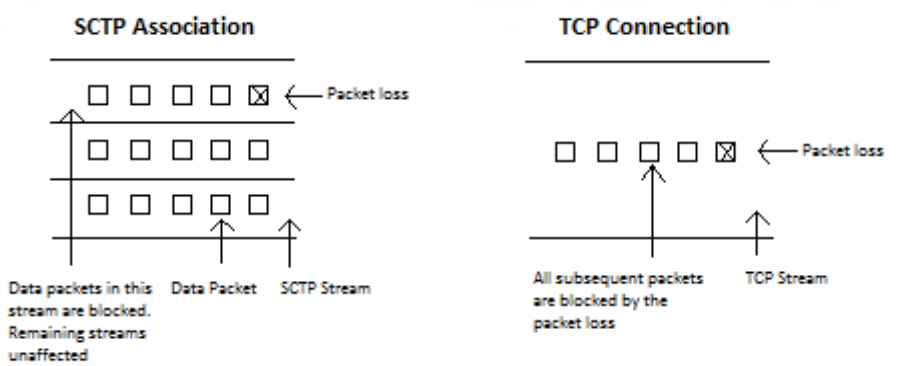


*Figure 3: This picture illustrates the differences between a TCP connection and an SCTP association. Redrawn from Iqbal (2003).*

### 2.2.2 Multi-homing

A TCP connection between two end-points is established between a single interface on each of the end-points. The end-points may have several interfaces and know of other routes to each other but if an interface or link fail the TCP connection would have to timeout, abort and force reestablishment of the connection on another interface (Amer & Stewart, 2007). SCTP introduces multi-homing which means that an association between two end-points can contain several interfaces at each end-point. The paths between these interfaces are monitored using a heartbeat mechanism, if a path failure is discovered SCTP can send the traffic over an alternate path seamlessly without application or user noticing it. Multi-homing is thus useful in cases where high availability is required and it increases the reliability of the association (Internet Engineering Task Force, 2007a).

# 3 Problem

There are weaknesses in the TCP protocol such as susceptibility to SYN-flooding and head-of-line blocking. As mentioned by Amer & Stewart (2007), Jones (2006) and others, SCTP was developed with these weaknesses in mind and handles things like session initiation and streams differently from TCP to solve them. In addition SCTP also introduces several new features which further increases the possible uses of the protocol.

Considering that there are known weaknesses in the TCP protocol and that SCTP alleviates or eliminates these weaknesses as well as supports several new features the aim of this thesis is to:

*Present proposed implementations of SCTP, according to available research, analyze them and issue recommendations on how critical it would be to move forward on these implementations.*

## 3.1 Boundaries

To keep the scope of the thesis reasonable there will be a limitation on the amount of implementations presented. Each new implementation will be addressing a unique weakness in the TCP protocol and/or take advantage of a unique major feature of SCTP. Implementations which address several weaknesses in TCP and/or takes advantage of multiple features of SCTP is likely to be presented but each new presentation will contain a weakness or feature which will not have been present in the preceding ones. This will effectively limit the amount of implementations presented as well as ensure a diverse selection of implementations.

## 3.2 Objectives

In order to fulfill the aim of the thesis it will be divided into several objectives which will need to be achieved. Some of these objective also has sub-questions which will have to be answered:

1. Identify suitable proposed implementations of SCTP.
2. Present the proposed implementations.
    1. What problem would the implementation solve?
    2. How would the implementation solve the problem?
3. Analyze the proposed implementations.
    1. Could the problem have been solved without using SCTP?
    2. What would the benefits be for implementing SCTP in this case?
    3. Is there any possible downsides with using SCTP to solve this problem?
4. Issue recommendations on how critical it would be to move forward on the proposed implementations.

# 4 Method

This chapter aims to explain how the objectives will be achieved and which methods will be used to achieve each objective. The primary method of this thesis will be to perform a literature analysis, this is defined by Berndtsson et al. (2008, p58) to mean:

*"a systematic examination of a problem, by means of an analysis of published sources, undertaken with a specific purpose in mind."*

Doing a literature analysis is well suited for this thesis since it is based on analyzing projects performed by different people in the area of SCTP with the purpose of issuing recommendations. The analysis will be systematic to the degree that it will follow the steps outlined in the rest of this chapter. The approach used will mostly be qualitative in its nature, since it would be difficult to put values on such things as fulfillment of criteria in a consistent way. Furthermore I feel that in this case it would offer no real benefits to do so since it is not a comparison between the different proposals that is the aim of this thesis but rather recommendations on the merit of each individual proposal.

## 4.1 Identifying suitable proposed implementations

The first step will be to identify suitable proposed implementations of SCTP to use in this thesis. To achieve this objective the primary activity will be searching through article databases for relevant research in the area of implementation of SCTP, identify suitable articles containing proposals for implementation and to make the selection of which to include in this paper.

To be a suitable proposal for implementation it will have to conform to the boundaries specified in section 3.1 as well as the following criteria:

- The implementation must solve a real problem which is clearly specified.
- The implementation must not be too general. A proposal to implement SCTP in order to protect a web server from SYN-flood attacks would be acceptable, a proposal to implement SCTP to stop SYN-flood attacks in general would not be acceptable.

These criteria will ensure that the selection will result in proposals for implementations which can be properly presented. In addition to this, and considering the boundaries of the selection, the chosen implementations will as far as possible be representative of the SCTP features they involve so that even if the amount of implementations presented here will be small they will be closely related to similar implementations.

## 4.2 Presenting the proposed implementations

The second step will be to present the proposed implementations that were identified and selected in step 1. The presentation of the individual proposals may vary but to analyze and issue recommendations on equal criteria they must all contain answers to the sub-questions supplied in section 3.2:

- What problem would the implementation solve or which features would the implementation take advantage off?
- How would the implementation solve the problem? For example which features of the SCTP protocol will be used to solve it?

## 4.3 Analyzing the proposed implementations

In the third step there will be an analysis of the proposed implementations presented in step 2. Once again the analysis of the individual proposals may vary but in order to standardize the analysis somewhat and in order to make issuing recommendations easier the following sub-questions from section 3.2 will be answered:

- Could the problem have been solved without using SCTP?
- What would the benefits be for implementing SCTP in this case?
- Is there any possible downsides with using SCTP to solve this problem?

The answer to these questions will found by performing a study of relevant literature.

## 4.4 Issuing recommendations

The final step will be to issue recommendations based on the analysis performed in step 3. To standardize the recommendations three different levels will be used: Critical, Recommended and Not Recommended. The criteria for each level is as follows:

- **Critical**: The implementation of SCTP in this case is not only recommended, it is urgent. There is no equivalent way of solving the problem without using SCTP and the consequences of not implementing SCTP are so severe as to warrant immediate action.
- **Recommended**: The implementation of SCTP in this case would provide a clear benefit. Either no equivalent way of solving the problem using another protocol could be found or they provide less benefits than SCTP.
- **Not Recommended**: The implementation of SCTP in this case would not provide a clear benefit, there might be an equivalent way of solving the problem using another protocol. In the worst cases implementing SCTP would provide harmful effects rather than benefits.

These criteria and the levels of recommendation were developed by the author of this paper since relevant and already existing standards to use could not be found. The use of three levels is because there is only three relevant recommendations to give: implement immediately (Critical), implement (Recommended) and do not implement (Not recommended). It is not relevant to further divide these levels and the critical level is needed to distinguish between those recommendations where implementation is urgent and those where it is not.

In addition to the recommendation itself, a clear justification will be given for why a certain recommendation was given.

# 5 Identification of suitable implementations

A study of the available research material regarding proposed implementations of SCTP led to the selection of the following implementations: HTTP over SCTP, Online Games over SCTP and IP Mobility over SCTP. This chapter will explain why each of these implementations were chosen.

**HTTP over SCTP**: This implementation makes use of several major features of SCTP to remedy the issues with using TCP for HTTP traffic. The implementation is as representative as can be for the features of multi-streaming and multi-homing as well as for the protection against SYN-flooding (Amer et al., 2006) which makes other implementations covering these features redundant for the purpose of this paper.

**Online Games over SCTP**: This proposed implementation shows how well SCTP handles thin-stream traffic which is common for online games. This is of interest because of the wide variety of applications which produces such traffic and since the retransmission and congestion control mechanisms of TCP are ill equipped for it (Petlund et al., 2009). Other SCTP features which could benefit online games, and which will be covered by this paper, are partial reliability and unordered data delivery (Amer et al., 2006).

**IP Mobility over SCTP**: This proposed implementation takes advantage of the way SCTP uses associations to keep track of IP addresses for an end-to-end connection (Kim & Koh, 2008). This implementation was chosen to show that there are other uses for SCTP than to act as a replacement for TCP.

These implementations cover a lot of the available features in SCTP and present uses for them that are representative for how they can be used in other implementations. Other reasons for choosing these three was the existence of relevant articles and research on the subjects as well as the fact that they all would have an impact on a lot of people. The web infrastructure is encountered by basically anyone who uses Internet, online games are played by millions of people across the globe and the increased usage of mobile devices such as laptops and mobile phones makes working IP mobility an important subject.

# 6 Results

This chapter contains the proposed implementations which have been chosen. First, each implementation is briefly described and then an analysis and recommendation on their importance follows, all according to the method outlined in chapter 4.

## 6.1 HTTP over SCTP

This implementation presents the idea of using SCTP as a transport protocol for the Hypertext Transfer Protocol (HTTP) infrastructure: web servers and web browsers. The choice of this implementation was based on the large number of SCTP features it would benefit from, as well as on how prevalent HTTP is in the current Internet infrastructure (Amer et al., 2006).

### 6.1.1 Presentation

HTTP is an application layer protocol which requires a reliable transport protocol for its end-to-end communication. TCP is the transport protocol commonly used to ensure the reliable transportation of HTTP traffic and has lacked any reasonable alternative. The development of SCTP offers such an alternative since it is also a reliable transport protocol which offers new services that are unavailable in TCP while rectifying several weaknesses with using HTTP over TCP (Rajamani et al., 2002).

According to Amer et al. (2006) there are three major concerns with using TCP as a transport protocol for TCP: head-of-line blocking, network failures and SYN-flood attacks.

#### 6.1.1.1 Head-of-line blocking

Head-of-line blocking occurs because TCP enforces strict ordering of data. When packets are lost and following packets arrive out of order they are held until the lost packet is retransmitted. Since all packets are sent over a single stream this leads to all data being delayed, not just the ones related to the lost packet, resulting in a delay for the end-user (Smith, 2009).

The effects of head-of-line blocking on HTTP traffic is especially notable in networks with low bandwidth and/or high packet loss rates. These characteristics are common for mobile devices such as laptops and mobile phones. In recent years the use of web browsers and other web applications for mobile devices have increased steadily which increases the importance of solving the problem with head-of-line blocking (Amer et al., 2006).

If HTTP traffic would use SCTP as a transport protocol it would benefit from the multi-streaming feature which, in turn, would decrease the effect of head-of-line blocking for the end-user. Instead of transmitting all packets over the same stream SCTP utilizes multiple streams and sends related packets over the same stream. There will still be delays for lost packets but only for the single stream in which the packet is lost, unrelated traffic will not be affected (Amer et al., 2006).

#### 6.1.1.2 Network failures

It is common for critical web servers to have multiple levels of redundancy in order to provide uninterrupted services. Having multiple network interfaces on a device could be such a redundancy. TCP can not properly make use of such an extra interface since the TCP connection established between a single interface on each end-point. In case of a link or interface failure the TCP connection would have to timeout, abort and force reestablishment of the connection on another interface (Amer & Stewart, 2007).

HTTP over SCTP would use the multi-homing feature of SCTP to take advantage of extra interfaces and links to provide an extra layer of redundancy. In SCTP an association is formed between the two end-points. This association has knowledge of all interfaces available on both end-points and monitors them with a heartbeat mechanism. If a path failure is detected the traffic can be sent over an alternate path seamlessly, neither applications or users will note the path failure (Amer et al., 2006).

#### 6.1.1.3 SYN-flood attacks

A SYN-flood occurs when an attacker takes advantage of a weakness in the three-way handshake for connection establishment in TCP. The attack is possible because the end-point being asked to establish a connection will allocate resources before the end-point asking for the connection. Unless the connection is fully opened by the initiating end-point there will be half-open connections which will fill up a buffer which keeps track of such. No new connections will be opened before these half-open connections time out effectively denying access to the end-point for legitimate users (Pfleeger & Pfleeger, 2006).

Denial-of-service attacks, such as SYN-flooding, can often deny access to the services provided by web servers, it is therefore desirable to limit the availability of such attacks. HTTP over SCTP would be completely protected from SYN-flooding (Amer et al., 2006). The attack has been eliminated by the use of a four-way handshake in which the end-point receiving a connection request does not open the connection until after the initiator have done so. Instead of saving information about the connection in a half-open state a cookie containing the information is sent to the initiator who must open the connection and then send back the cookie to fully open the connection (Internet Engineering Task Force, 2007a).

### 6.1.2 Analysis

In this section the problems and solutions for HTTP over SCTP is analyzed based on the sub-questions from section 4.3.

#### 6.1.2.1 Head-of-line blocking

As mentioned in section 2.2.1, it is possible to partially avoid the issue of head-of-line blocking while using TCP by allowing the web browser to open multiple TCP connections to a web server and distribute the HTTP data requests between them. This method is in frequent use today in the web infrastructure but has several flaws (Amer et al., 2006).

**Congestion behavior:** TCP connections are based on the principle of fairness when it comes to congestion. If a single applications opens multiple TCP connections and

there is congestion in the network the application will use more than its fair share of the bandwidth at the detriment of other applications and users sharing the medium (Iqbal, 2003).

**Increased load:** A web server has to allocate a Transmission Control Block (TCB) for each individual connection to keep track of it. Parallel TCP connections from a client to a server would increase the processing load of the server. One consequence of such high loads is that the server could drop new TCP requests due to a lack of resources (Amer et al., 2006).

**Connection establishment:** Each individual TCP connection would still have to go through the three-way handshake of the protocol before transmitting any data. This connection establishment for multiple connections is extra traffic on the network. Furthermore a packet loss during the connection establishment is time consuming since the only mechanism that can trigger a retransmission during it is a timeout (normally retransmissions occur when subsequent packets arrive with ACK-numbers which are out of order). This often results in an increase of the overall transfer time of data (Amer et al., 2006).

If these flaws did not exist the use of multiple TCP connections to counter the head-of-line blocking issue would be a viable comparison to using SCTP, as it stands the problem with head-of-line blocking in HTTP traffic cannot be solved in a satisfactory manner without the use of SCTP.

Amer et al. (2006) have tested an implementation of HTTP over SCTP using a modified Apache web server and a modified Firefox web client to demonstrate the effect of SCTP on head-of-line blocking for traffic over a low bandwidth connection (56kbps) with artificially induced packet loss (10% loss from server to client). Identical data was sent using TCP and SCTP and the same packets were lost. The result of the experiment clearly showed the delays in delivery of data which was unrelated to the lost packets for the TCP connection, on occasion several seconds, whereas the SCTP association delivered unrelated data instantly on arrival (see Table 1).

*Table 1: This table shows the results of a test run by Amer et al. (2006). Object 2 and object 4 suffered packet loss while the remaining objects did not.*

| | Delivery of object (TCP) | Delivery of object (SCTP) | Difference |
|---|---|---|---|
| Object 1 | 0.8 seconds | 0.8 seconds | 0 seconds |
| Object 2 | 4.2 seconds | 4.2 seconds | 0 seconds |
| Object 3 | 4.2 seconds | 2.4 seconds | 1.8 seconds |
| Object 4 | 4.5 seconds | 4.5 seconds | 0 seconds |
| Object 5 | 4.5 seconds | 4.0 seconds | 0.5 seconds |

Implementing HTTP over SCTP would not have any large effects for end-points in networks with high bandwidth and low loss rates. End-points located in networks with low bandwidth and high loss rates would however experience much less negative effects from head-of-line blocking. The improvements for the latter would be very noticeable if HTTP over SCTP was implemented.

#### 6.1.2.2 Network failures

When it comes to transport protocols and multi-homing it is quite straightforward. SCTP is the only transport protocol which has support for multi-homing. It is not possible to achieve true multi-homing with any other protocol. (Amer & Stewart, 2007).

Implementing HTTP over SCTP would have the largest impact for really critical web servers where it would serve as an extra layer of redundancy and remove the weakness of having a single point of failure. It is possible to use in other cases as well and there have been suggestions to use the multi-homing feature to create Concurrent Multipath Transfer (CMT) which would allow the sending of data over multiple end-to-end paths simultaneously through the same association (Janardhan et al., 2004).

#### 6.1.2.3 SYN-flood attacks

There are several ways to mitigate the effect of SYN-flood attacks and to partially prevent them. These include: IP based filtering, increasing backlog, reducing SYN-RECEIVED timer, recycling the oldest half-open TCB, SYN Cache, SYN Cookies, firewalls and proxies (Internet Engineering Task Force, 2007b). This paper does not discuss these solutions in detail but it is noted that none of them can fully stop a SYN-flood attack and that most of them come with their own drawbacks and the need to manually configure either every exposed web server or its firewall to deal with SYN-flood attacks. There is as of yet no solution to the SYN-flood attack other than the implementation of SCTP which completely removes the problem.

Implementing HTTP over SCTP would completely remove the threat of SYN-flood attacks on web servers which would deny access to them for legitimate users. It would also remove the need to manually configure web servers and firewalls to mitigate the effect of such attacks (Amer et al., 2006).

#### 6.1.2.4 Other considerations

Since SCTP was designed to solve these specific issues with TCP it is no surprise that there are few specific downsides with using SCTP in these cases. Any possible downsides are more general, such as the fact that SCTP is a very young protocol which has not had the decades of widespread use of TCP and could have yet undetected weaknesses. For example the TCP vulnerability to SYN-flood attacks was detected as late as 1994, after more than a full decade of heavy use of the protocol (Thorstensson. 2010).

### 6.1.3 Recommendation

The proposed implementation for HTTP over SCTP highlights several issues with the TCP implementation all of which can be solved by implementing SCTP. The analysis has found no satisfactory solutions other than the implementation of HTTP over SCTP to solve these issues. The benefits of implementing SCTP are however not large enough for it to be urgent to start implementing this solution. The recommendation for HTTP over SCTP is for these reasons:

***Recommended****: The implementation of SCTP in this case would provide a clear benefit. Either no equivalent way of solving the problem using another protocol could be found or they provide less benefits than SCTP.*

## 6.2 Online Games over SCTP

This implementation presents the idea of using SCTP as a transport protocol for online games to reduce latency and take advantage of some additional features of SCTP. The thought of using SCTP as a transport protocol for online games comes from the fact that the traffic of such games is similar to the signaling traffic which SCTP was originally developed to handle. The idea behind choosing to present this particular implementation is that the conclusions drawn can be applied to applications which handle similar types of traffic. Examples of such traffic include stock trading, thin clients, control systems, remote probe operations and audio conferencing (Petlund et al., 2009).

### 6.2.1 Presentation

Online games, in general, provide services which rely on the transport protocol to deliver low latency and timely arrival of data. Anything other than consistent low latency degrades the interactive experience of the user with the limits of player tolerance ranging from 100ms for First Person Shooters and Racing games up to 1000ms for Real Time Strategy games with MMORPGs (Massively Multiplayer Online Role Playing Games) somewhere in the middle with 500ms (Claypool & Claypool, 2006).

In addition to the hope of improving latency Amer et al. (2006) have identified two features of SCTP that online games could benefit from: partial reliability and unordered data delivery.

#### 6.2.1.1 Latency

Using TCP is common for online games since there are demands on reliable arrival of data. The most famous example is World of Warcraft with over 12 million subscribers (Blizzard Entertainment, Inc., 2010). The methods used for congestion control and retransmission of lost packets in TCP can however lead to high latency, especially considering that these are not optimized for thin streams. Thin streams are characterized by having small packets and high packet inter-arrival times (Petlund et al., 2009).

Congestion control exists in order to ensure fairness amongst applications using the same bandwidth and governs the behavior of both TCP and SCTP. The algorithms that ensures congestion control are optimized for greedy applications which sends data often. One of the effects of congestion control is that the timeout value is doubled for each new timeout retransmission of a lost packet (Petlund et al., 2009).

In general, there are two ways in which a retransmission of a lost packet gets triggered in TCP. The first is if enough subsequent packets (three) has arrived at the destination and the sender has gotten the ACKs for those packets, indicating that the the preceding packet has been lost. The second way to trigger a retransmission is with a timeout, if enough time has passed since the packet was sent it will be deemed lost and a retransmission will occur. The time it takes for a timeout to occur is based on Round-Trip Time (RTT), the time it takes from sending a packet until the ACK has been received, but since the traffic of online games tend to have high inter-arrival

time it is very common that a timeout triggers the retransmission of a lost packet (see Figure 4) (Petlund et al., 2009).

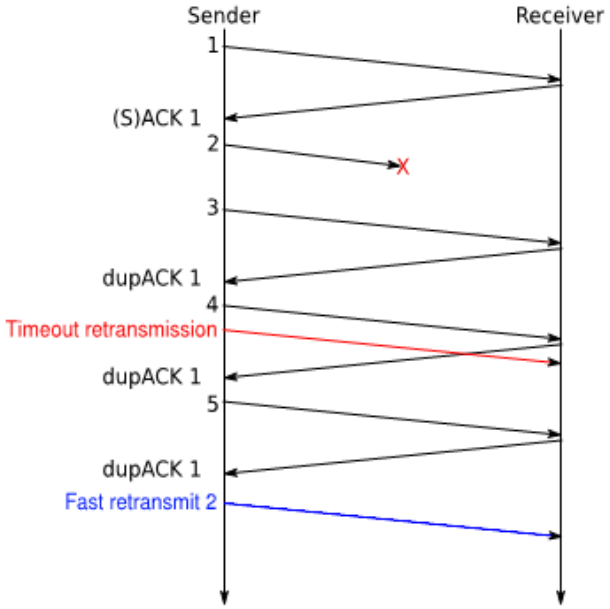


*Figure 4: This picture shows how a fast retransmit will not occur in a thin stream because not enough packets are sent before a timeout is triggered (Petlund et al., 2009. Picture released under Creative Commons Attribution Noncommercial License)*

SCTP handles retransmissions in much the same way as TCP does. If enough subsequent packets in a stream arrives after a packet has been lost a retransmission is triggered. Timeouts are also handled the same way with the same consequences for traffic with high inter-arrival time. One additional negative aspect with SCTP is that the minimum value for timeout is set to 1000ms by default whereas the same setting for TCP is only 200ms. One strength of SCTP is that it is message oriented, it keeps track of the messages contained in a packet rather than the packet itself, when a packet is lost the messages from it can be bundled together with other outgoing messages in a new packet instead of retransmitting the original packet (Harcsik et al., 2008).

Considering the high inter-arrival times of traffic for online games, especially the minimum value of the timeout, the standard variant of SCTP seems ill suited to act as a transport protocol for such applications. Petlund et al. (2009) and Harcsik et al. (2008) have made experimental studies which collaborates this but they have also identified several modifications that could be made to SCTP which would drastically improve its performance. These improvements include reduced minimum timeout value, reduced amount of subsequent messages to trigger a retransmission and reducing the modifier for subsequent retransmissions (from doubled each retransmission to a more linear progression). These changes would allow a greedy

application to use an unfair share of the bandwidth however, therefore a thin stream detector is also included. These modifications will only be used whenever an application is deemed to have thin stream characteristics.

#### 6.2.1.2 Partial Reliability

With TCP all data is sent with full reliability, this means that any lost packet will be retransmitted until it reaches its destination. SCTP, on the other hand, has support for partial reliability which gives applications flexibility in how strictly reliability should be enforced for retransmitting lost packets. An application can specify a lifetime for a message and when the lifetime expires the message is removed regardless of if the message has been successfully transmitted. This feature is useful when the application constantly generates new data which obsoletes the old one. An example for online games would be a player moving around and constantly generating new coordinates for its position (Amer et al., 2006).

#### 6.2.1.3 Unordered data delivery

TCP enforces strict ordering for all data which means that if both ordered and unordered data is sent over a TCP connection there will be a delay in delivering the unordered data to the application, this is related to head-of-line blocking since packet loss will keep unordered data waiting unnecessarily. With SCTP an application can mark a message for unordered delivery which makes SCTP hand over the packet to the application instantly upon arrival. This feature could for example be useful for games which wants to implement an in-game voice-chat (Amer et al., 2006).

### 6.2.2 Analysis

In this section the problems and solutions for online games over SCTP is analyzed based on the sub-questions from section 4.3.

#### 6.2.2.1 Latency

One alternative to TCP and SCTP is to send online game traffic over UDP, this typically results in lower latency but requires the application itself to handle such matters as reliability, congestion control and ordering of data. Generally, UDP is chosen for online games which has the most stringent demands on latency, such as FPS (First Person Shooter) and racing games. It does, however, introduce the need to process the information through the application rather than on the transport layer. This means added strain on the computer running the application. There is also cases when firewalls block UDP traffic, normally online games based on UDP will use TCP as a fallback in those cases (Harcsik et al., 2008).

Similar enhancements to those proposed for SCTP has also been proposed for TCP by Petlund et al. (2008) with results that show large improvements over the default settings and could be used to reduce latency in online games.

While the standard variant of SCTP results in unacceptably high latency, the modified implementation shows better results which matches or outperforms the common variants of TCP as well as the enhanced one in tests performed by Harcsik et al. (2008) and Petlund et al. (2009).

Implementing the standard variant of SCTP would not improve latencies and in many cases have a deleterious affect on them. There are alternatives which perform better latency-wise for online games such as TCP and UDP. Implementing the modified variant of SCTP does however compare favorably with TCP regarding latency and can compete even with UDP while not having the same negative effects, it is however experimental and needs further research (Petlund et al., 2009).

#### 6.2.2.2 Partial Reliability

It is possible to implement partial reliability to a certain extent in the application layer while using UDP. The same issues that are presented in section 6.2.1.1 for UDP are present here as well though: added strain on the computer and firewalls blocking UDP (Harcsik et al., 2008).

Implementing SCTP for online games would provide the benefits of partial reliability without the extra strain caused by implementing UDP with application-based reliability-handling.

#### 6.2.2.3 Unordered data delivery

UDP also allows for unordered data delivery but provisions for ordered data must in that case be implement in the application layer for it to be used for online games (Petlund et al., 2009). There are also cases of online games which use TCP for ordered data and UDP for unordered data, one such example is Starcraft II which uses TCP to play the game and UDP for its voice-chat (Blizzard Entertainment, inc. 2011). The same issues from previous sections are still valid though, pure UDP strains the computer and firewalls blocking UDP causes issues for both UDP and TCP+UDP.

Implementing SCTP would provide the benefits of unordered data delivery while still being able to send ordered data and avoids the issues with using UDP.

### 6.2.3 Recommendation

The alternatives to SCTP are far from ideal for handling traffic from online games, however the standard implementation of SCTP fails to meet the requirements when it comes to latency. A modified implementation of SCTP would seemingly meet those demands (Petlund et al., 2009) but it is in an experimental phase and would need further testing to see that it does not cause congestion issues when used by applications which use more bandwidth.

Latency is the primary concern for online games and the benefits of partial reliability and unordered data delivery does not outweigh it. There are available protocols which provide better latency than SCTP. For these reasons I cannot recommend that the standard variant of SCTP is implemented for online games. My recommendation for online games over SCTP is therefore:

***Not Recommended****: The implementation of SCTP in this case would not provide a clear benefit, there might be an equivalent way of solving the problem using another protocol. In the worst cases implementing SCTP would provide harmful effects rather than benefits.*

## 6.3 IP mobility over SCTP

Mobile IP is a network layer standard which provides seamless mobility for a mobile node when moving between different networks. This implementation presents the idea of using SCTP to perform IP handover on the transport layer to take advantage of the way SCTP supports the use of multiple IP addresses within one association. This is hoped to lower the latency for the handover process which consists of many steps in mobile IP (Kim & Koh, 2008).

### 6.3.1 Presentation

Mobile IPv6 (MIPv6) is the latest proposed standard for mobile IP and is an integrated part of the IPv6 protocol, every device which supports IPv6 also has built-in support for MIPv6 whereas MIPv4 is an extension-protocol which must be added on all devices which wants to use it (Kim & Koh, 2008).

The basic operation of MIPv6 is that a home agent provides location management for the mobile nodes. A mobile node moves between access points on different networks but is reachable on its home address through the home agent which keeps track of the current location of the mobile node. The mobile node establishes connections to other nodes through the home agent in order to ensure that the connection can be maintained when the mobile node moves to a new network. When a mobile node attaches itself to a new access point it is provided with a new IP address which can be used to direct communication between the nodes after the connection has been established through the home agent (Kim & Koh, 2008).

*Table 2: A comparison of Mobile IPv6 and Mobile SCTP*

| | Mobile IPv6 | Mobile SCTP |
|---|---|---|
| Protocol layer | Network layer | Transport layer |
| Location management | Provided | Not provided |
| Mobility agents | Home Agent | No need of mobility agents |
| Handover support | With the help of the mobility agents | Provided |

Table 2 includes a comparison of MIPv6 and mobile SCTP. It shows that mobile SCTP does not have any support for location management but provides handover support without needing to go through the home agent. It is possible to use MIPv6 for location management and SCTP for the handover process (Kim & Koh, 2008).

#### 6.3.1.1 Handover

A handover occurs when a mobile node moves from one network to another while having active connections to other nodes. The mobile node must inform these nodes of its new location in order to maintain the connection, any packets sent during this

process are lost and must be resent after it is completed, it is therefore important that the process is completed as swiftly as possible (Kim & Koh, 2008).

The handover process for MIPv6 includes several steps which contributes to delay: (1) the mobile node detects movement into a new network, (2) the mobile node configures the new IP address which it receives from the new access point it attaches itself to, (3) the new address is registered with the home agent, (4) finally the new address is registered with the nodes with which the mobile node has active connections with (Kim & Koh, 2008).

SCTP handles the handover process in a completely different fashion, since all the changes occur in the association between the mobile node and other nodes. There is no need for the other nodes to confirm the changed IP address with a home agent. Instead the SCTP handover consists of the following steps: (1) obtain IP address from a newly discovered network, (2) add the new IP address to the active SCTP associations, (3) change the primary IP address used by the associations, (4) delete the old IP address from the associations (Koh et al., 2004).

### 6.3.2 Analysis

In this section the problems and solutions for IP mobility over SCTP is analyzed based on the sub-questions from section 4.3.

#### 6.3.2.1 Handover

MIPv6 performs handover in the network layer while SCTP does it in the transport layer. A third option would be to perform handover in the application level by using Session Initiation Protocol (SIP) which is an application-layer protocol for establishing, modifying and terminating multimedia sessions. It does, however, require a functioning SIP infrastructure with SIP servers in order to function. While SIP outperforms MIPv6 in the delay incurred by the handover process it performs worse then SCTP (Zeadally & Siddiqui, 2007)

The main issue with the handover process is the delay which is incurred. Since MIPv6 needs to register any changes with the home agent in order to register them with other nodes an extra delay is introduced which is not present when using SCTP. In an experiment done by Kim & Koh (2008) the average handover latency for SCTP was 2.077s while MIPv6 had an average handover latency of 4.188s. This matches the tests by Zeadally & Siddiqui (2007) which showed a 55% lower average handover latency for SCTP compared to TCP.

Implementing SCTP would yield much lower delays when performing a handover for mobile devices than if MIPv6 were to be used. While SIP would also be a possible candidate for IP mobility, it is noted that an existing SIP infrastructure would have to be present and, even if that would be the case, SCTP still outperforms SIP when it comes to handover delay.

### 6.3.3 Recommendation

The currently proposed standard for IP mobility (MIPv6) incurs a significant delay during the handover process. Using SCTP for the handover process would yield much lower delays without having any adverse affects. Therefore the recommendation for IP mobility over SCTP is:

***Recommended**: The implementation of SCTP in this case would provide a clear benefit. Either no equivalent way of solving the problem using another protocol could be found or they provide less benefits than SCTP.*

## 6.4 Analysis of results

The analysis of the three proposed implementations presented in this chapter has led to some interesting results. The most surprising was probably the poor performance of the current implementations of SCTP for latency in online games. Since the signaling traffic which SCTP was originally developed to handle is very similar to the traffic produced by online games it is surprising that the default settings for retransmission in SCTP are such that they lead to large delays for such traffic. The modifications proposed by Petlund et al. (2009) does, however, solve these issues and they state that it is easy to distinguish between thin streams and thick streams. If that is the case, their proposal for SCTP to consider applications with thin streams and thick streams separately and adjust accordingly would seem to have merit, if such changes were made to SCTP it would make it a more likely candidate for online games. As it is the current implementation of SCTP cannot be used for online games without resulting in high latency, both Petlund et al. (2009) and Harcsik et al. (2008) agree on this point and that the current implementations of SCTP should not be used for online games.

Amer et al. (2006) argues that the additional features of SCTP make it a better protocol for the web infrastructure than TCP. In particular, the negative effect of head-of-line blocking on mobile devices suffering from more packet loss is pointed out. The analysis of HTTP over SCTP performed in this paper points in the same direction which is unsurprising since much of the analysis is based on the work carried out by of Amer et al. (2006). Other works, such at Natarajan er al. (2009) and Rajamani er al. (2002), seem to corroborate this view by presenting results from experiments and arguments for why SCTP is a better transport protocol for HTTP traffic. No research indicating anything else was found.

Finally it is interesting that IP mobility over SCTP is another implementation which affects mobile devices in a favorable manner by using SCTP. Kim & Koh (2008), Zeadally & Siddiqui (2007) as well as Koh et al. (2004) all agree that the features of SCTP make it an excellent candidate to handle handover for mobile devices that move from network to network. It seems quite possible that the features of SCTP make it a friendly protocol for mobile devices.

# 7 Conclusions

The aim of this thesis was to: "*present proposed implementations of SCTP, according to available research, analyze them and issue recommendations on how critical it would be to move forward on these implementations."* In section 3.2 several objectives were specified as well as several sub-questions that needed to be answered in order to arrive at a recommendation were established. In Table 3 a summary of the results can be seen, the table includes the problems the implementation would solve, how the implementation would solve the problems and what recommendation was issued for the implementation.

*Table 3: This table summarizes the results of this paper.*

| **Implementation:** | **Problems:** | **Solutions:** | **Recommendation** |
|---|---|---|---|
| HTTP over SCTP | Head-of-line blocking<br>Network failures<br>SYN-flood attacks | Multi-streaming<br>Multi-homing<br>Four-way handshake | Recommended |
| Online Games over SCTP | High latency | Not solved | Not recommended |
| IP Mobility over SCTP | High handover latency | SCTP association | Recommended |

The analysis of HTTP over SCTP showed clear gains from using SCTP as the transport protocol for web traffic. This result should be of interest for organizations with web servers that are critical and that are subject to DDoS attacks, needs an extra layer of redundancy and/or which services are commonly used by mobile devices. Such web servers would benefit the most from the features provided by SCTP.

Even though online games over SCTP was not recommended the analysis showed that the implementation can be of interest if the correct changes can be made. Online games could benefit from SCTP features such as partial reliability and unordered data without having to implement them on the application layer.

The results show that there are instances when SCTP would be preferable to use instead of the more commonly used TCP. They also indicate that there are still some issues with SCTP which make it unsuitable for certain applications. A conclusion to draw from this would be that even though there are uses for SCTP it is not yet ready to fully replace TCP as the transport protocol of choice. It is, however, important to continue the research of SCTP with the aim of making our Internet infrastructure more robust.

The fact that no implementation reached the recommendation level of Critical indicates that even though there are some weaknesses associated with the use of TCP they are not severe enough to warrant urgent action.

# 8 Reflections

During the writing of this thesis, no major difficulties were encountered other than finding relevant material to compare results against. Since SCTP is a relatively new field of research, the available material consists in large part of proposed implementations of different kinds. These proposals are often very good at highlighting the positive features of SCTP but lack criticism and finding possible downsides. Another large part of the available material consists of experiments which present results but which does not issue any recommendations from them. This made it hard to match the issued recommendations against related works since they did not contain any such recommendations.

There were other possible implementations that could have been included in this report, for example Diameter over SCTP, which were discarded due to lack of research material of high enough quality. The three proposed implementations chosen for this thesis did have the highest quality of research out of the candidates, especially the articles regarding online games over SCTP held high standard and approached the implementation in a critical fashion.

Following the method, constructed in chapter 4, made it relatively easy to structure the thesis. By answering the sub-questions and then building on those answered the objectives could be met while maintaining coherency. The only major weakness of the method is that it relies entirely on the works of others. Writing a thesis with more practical elements would probably have furthered the field of research more but it was deemed that the scope of the course did not allow for the time that most practical experiments would consume.

It is the hope of the author that this thesis will inspire further work regarding the implementation of SCTP. It is a field of research with great potential for the future development of the Internet infrastructure.

## 8.1 Future work

A lot of research has been performed regarding SCTP since the protocol was introduced. There is, however, still a lot of work to be done in this field. Regarding the implementations covered in this thesis the following could be considered for future work.

**HTTP over SCTP:** Most of the research in this area has focused on determining SCTP:s suitability as a transport protocol for web traffic. The recommendation in this thesis is to move forward with the implementation of SCTP, this also seems to be the general consensus amongst the literature on the subject. There is, however, one area pertaining to the subject which has not seen much research but which would have to be addressed before SCTP could be implemented on wide scale. How should the transition from TCP to SCTP be handled? The transition would be far from instantaneous, web servers using SCTP would coexist with web servers using TCP, or both, for a long time. The most important issue to study is how the client would decide which protocol to use for HTTP traffic since different web servers would be using different protocols. Possible solutions to test would be having a primary

protocol to try first and then use a secondary one if the first one fails or to have the client and the server arbitrate the protocol to use between them. Testing different solutions and measuring the delay involved in them would be of great interest.

**Online Games over SCTP:** The research performed by Petlund et al. (2009), Harcsik et al. (2008) and others at Simula Research Laboratory and University of Oslo has been exhaustive and produced results which show that the current implementations of SCTP are not suitable for thin-stream traffic. Several improvements to the current implementations have been proposed and tested for thin-stream traffic. These changes would make SCTP too greedy to be used with applications that does not produce thin-stream traffic. An example of how future work would be to develop and test algorithms that determines if traffic is thin-stream or not. The implementation of such algorithm would provide the best of two worlds, aggressive retransmission behavior for thin-stream applications while being able to maintain congestion control for more greedy applications.

**IP Mobility over SCTP**: Considering the results of this thesis, that both HTTP over SCTP and IP mobility over SCTP were recommended and have positive effects on mobile devices, it would be interesting to see a study on the overall effects on mobile devices if SCTP were to replace TCP as the standard protocol for reliable traffic.

# References

Amer, P. Iyengar, J. Natarajan, P. & Stewart, R. (2006) *SCTP: An Innovative Transport Layer Protocol for The Web,* 15th International conference on World Wide Web. Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.146.2166 [Accessed 11.04.21]

Amer, P. & Stewart, S (2007) *Why is SCTP needed given TCP and UDP are widely available,* Internet Society. Available online: http://www.isoc.org/briefings/017/ [Accessed 11.02.22]

Berndtsson, M. Hansson, J. Olsson, B. & Lundell, B (2008) *Thesis Projects – A Guide for Students in Computer Science and Information Systems* (2nd edition). London: Springer Verlag.

Blizzard Entertainment, Inc. (2010) *WORLD OF WARCRAFT® SUBSCRIBER BASE REACHES 12 MILLION WORLDWIDE.* Available online: http://us.blizzard.com/en-us/company/press/pressreleases.html?101007 [Accessed 11.04.20]

Blizzard Entertainment, Inc. (2011) *Starcraft II Port Information – Firewalls, Routers, and Proxies.* Available online: http://us.blizzard.com/support/article.xml?locale=en_US&articleId=34063 [Accessed 11.04.20]

Claypool, M. & Claypool, K. (2006) *Latency and player actions in online games.* Available online: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.98.330 [Accessed 11.04.20]

Fu, S. & Atiquzzaman, M. (2004) *SCTP: State of the art in Research, Products and Technical Challenges.* Available online: http://www.cs.ou.edu/~atiq/papers/03-CCW-sctp-state-art.pdf [Accessed 11.02.22]

Harcsik, S. Petlund, A. Gridwodz, C. Halvorsen, P. Simula Research Laboratory, Norway. & IFI, University of Oslo, Norway. (2008) *Latency Evaluation of Networking Mechanisms for Game Traffic.* Available online: http://heim.ifi.uio.no/~paalh/publications/files/netgames2007.pdf [Accessed 11.04.22]

Internet Engineering Task Force (1981) *Transmission Control Protocol,* Request For Comments 793. Available online: http://tools.ietf.org/rfc/rfc0793.txt [Accessed 11.02.23]

Internet Engineering Task Force (2002) *Stream Control Transmission Protocol Applicability Statement,* Request For Comments 3257. Available online: http://www.ietf.org/rfc/rfc3257.txt?number=3257 [Accessed 11.02.18]

Internet Engineering Task Force (2007a) *Stream Control Transmission Protocol,* Request For Comments 4960. Available online: http://www.rfc-editor.org/rfc/rfc4960.txt [Accessed 11.02.18]

Internet Engineering Task Force (2007b) *TCP SYN Flooding Attacks and Common Mitigations,* Request For Comments 4987. Available online: http://tools.ietf.org/html/rfc4987 [Accessed 11.02.18]

Iqbal, A. (2003) *SCTP Primer.* Imperial College, London, UK. Available online: http://datatag.web.cern.ch/datatag/WP3/sctp/primer.htm [Accessed 11.02.23]

Janardhan, R. Iyengar, J. Keyur, S. & Amer, P. (2004) *Concurrent Multipath Transfer Using SCTP Multihoming.* Available Online: http://www.cis.udel.edu/~amer/PEL/poc/pdf/TR2004-02.CMT.Iyengar.pdf [Accessed 11.04.21]

Jones, T. (2006) *Better networking with SCTP.* developersWorks, IBM. Available online: http://www.ibm.com/developerworks/linux/library/l-sctp/ [Accessed 11.02.26]

Kim, D. & Koh, S. (2008) *Analysis of Handover Latency for Mobile Ipv6 and mSCTP.* Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.159.6649 [Accessed 11.04.24]

Koh, S. Chang, M. & Lee, M. (2004) *mSCTP for Soft Handover in Transport Layer.* Available online: http://155.230.105.164/pub/2004-cl.pdf [Accessed 11.04.23]

Labovitz, C. McPherson, D. Iekel-Johnson, S. & Hollyman, M. (2008) *Internet Traffic Trends – A View from 67 ISPs.* Available online: http://www.nanog.org/meetings/nanog43/presentations/Labovitz_internetstats_N43.pdf [Accessed 11.05.19]

Natarajan, P. Amer, P. Leighton, J. & Baker, F. (2009) *Using SCTP as a Transport Layer Protocol for HTTP.* Available online: http://tools.ietf.org/html/draft-natarajan-http-over-sctp-02 [Accessed 11.04.20]

Petlund, A. Beskow, P. Pedersen, J. Søgård Paaby, E. Griwodz, C. & Halvorsen, P. (2009) *Improving SCTP retransmission delays for time-dependent thin streams.* Available online: http://folk.uio.no/paalh/publications/files/mtap09-SCTP.pdf [Accessed 11.04.14]

Petlund, A. Evensen, K. Gridwodz, C. Halvorsen, P. Simula Research Laboratory, Norway. & Department of Informatics, University of Oslo, Norway. (2008) *TCP Enhancements for Interactive thin stream Applications.* Available online: http://simula.no/research/nd/publications/Simula.ND.112/ [Accessed: 11.04.21]

Pfleeger, C. P. & Pfleeger, S. L. (2006) *Security in computing* (4th edition). Upper Saddle River, New Jersey, USA: Prentice Hall.

Rajamani, R. Kumar, S. & Gupta, N. (2002) *SCTP versus TCP: Comparing the Performance of Transport Protocols for Web Traffic.* Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.120.2998 [Accessed 11.04.13]

Smith, T. (2009) *Exploring Head-of-Line Blocking.* Available online: http://document-program.blogspot.com/2009/11/105-exploring-head-of-line-blocking.html [Accessed 11.02.25]

Thorstensson J. (2010) *Genomgång av skyddsmetoder för TCP SYN flooding,* Högskolan I Skövde. Available online: http://his.diva-portal.org/smash/record.jsf?searchId=1&pid=diva2:323964 [Accessed 11.02.18]

Zeadally, S. & Siddiqui, F. (2007) *An empirical Analysis of Handoff performance for SIP, Mobile IP, and SCTP Protocols.* Available online: http://www.springerlink.com.persefone.his.se/content/a837lw8t34978572/ [Accessed 11.04.23]

# Appendix A – Acronyms

This appendix contains an alphabetical list of all the acronyms used in this thesis.

**CMT** – Concurrent Multipath Transfer

**DDoS** – Distributed Denial-of-Service

**HTTP** – Hypertext Transfer Protocol

**ICMP** – Internet Control Message Protocol

**IP** – Internet Protocol

**MIPv6** – Mobile Ipv6

**MMORPG** – Massively Multiplayer Online Role Playing Game

**RFC** – Request For Comments

**RTS** – Real Time Strategy

**SCTP** – Stream Control Transmission Protocol

**SIP** – Session Initiation Protocol

**TCP** – Transmission Control Protocol

**UDP** – User Datagram Protocol